GB04/4510

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed Stephen Hordley

Dated 4 November 2004

Patents Form 1/77

Patents Act 1977
(Rule 16)

THE PATENT OFFICE
A
- 5 NOV 2003
NEWPORT

The Patent Office

03NOV03 E849592-4 D02776
P01/7700 0.00-0325767.2

1/77

# Request for grant of a patent

*(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)*

The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

- 5 NOV 2003

1. Your reference — IP/P7305

2. Patent application number *(The Patent Office will fill in this part)* — 0325767.2

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

QINETIQ LIMITED

Registered Office 85 Buckingham Gate
London SW1E 6PD
United Kingdom

Patents ADP number *(if you know it)*

If the applicant is a corporate body, give the country/state of its incorporation — GB

8527376002

4. Title of the invention — Detection of Unwanted Items Stored in a Computer System

5. Name of your agent *(if you have one)*
"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Arthur Wyn Spencer WILLIAMS
QINETIQ LIMITED
IP Formalities
A4 Bldg
Cody Technology Park
Ively Road
Farnborough
Hants GU14 0LX United Kingdom

8183873001

Patents ADP number *(if you know it)*

| | Country | Priority application number *(if you know it)* | Date of filing *(day / month / year)* |
|---|---|---|---|
| 6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number | | | |

| | Number or earlier application | Date of filing *(day / month / year)* |
|---|---|---|
| 7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application | | |

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
   *a) any applicant named in part 3 is not an inventor, or*
   *b) there is an inventor who is not named as an applicant, or*
   *c) any named applicant is a corporate body. See note (d))*

Yes (b)



Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

| | |
|---|---|
| Continuation sheets of this form | |
| Description | 9 |
| Claim(s) | 5 |
| Abstract | 1 |
| Drawing(s) | 2 + 2 |

CF

~~10. If you are also filing any of the following,~~ state how many against each item.

| | |
|---|---|
| Priority documents | 0 |
| Translations of priority documents | 0 |
| Statement of inventorship and right to grant of a patent *(Patents Form 7/77)* | 2 |
| Request for preliminary examination and search *(Patents Form 9/77)* | 1 |
| Request for substantive examination *(Patents Form 10/77)* | 0 |
| Any other documents *(please specify)* | 0 |

11. I / We request the grant of a patent on the basis of this application.

Signature A W S WILLIAMS Date 04/11/2003

12. Name and daytime telephone number of person to contact in the United Kingdom: Mrs Linda Bruckshaw 01252 392722

**Warning**

*After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent of the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.*

**Notes**

*a) If you need help to fill in this form or have any questions, please contact the Patent Office on 0645 500505.*

*b) Write your answers in capital letters using black ink or you may type them.*

*c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

*d) If you have attached 'Yes' Patents Form 7/77 will need to be filed.*

*e) Once you have filled in the form you must remember to sign and date it.*

*f) For details of the fee and ways to pay please contact the Patent Office.*

# Detection of Unwanted Items Stored in a Computer System

This invention relates to detection of unwanted items stored in a computer system, and particularly, but not exclusively, to detection of items introduced by steganography.

Steganography may be defined as the covert concealment of information in the form of unwanted computer code within data on a carrier file such as an image file. The intention is that the existence of such information cannot be detected without some further information which is secret. Steganography is different to Cryptography. Cryptography can be described as being concerned with encrypting information so that an attacker cannot decipher it without some secret knowledge that it is hoped the attacker does not possess. Cryptography is therefore not necessarily concerned with secrecy of the existence of the message.

Steganography will normally be implemented with a large carrier file, so that only relatively minor perturbations are created in the carrier file by the introduction of secret information. These perturbations are small compared with the carrier document's apparent randomness. Many existing designers of steganographic systems use image files because of their large size. Known techniques include changing the least significant bit (LSB) of bitmap files or changing the LSB of some coefficients of JPEG files.

LSB changes are insignificant to human eyes, and so visual inspection will fail to detect steganographic information if the steganographic process is well designed.

However, LSBs are not truly random and they should show some statistical properties. Conceptually, a designer of a steganographic process may adopt some defensive techniques to avoid detection. The designer may compress data for input to the steganographic process, which tends to decrease the size of data to be hidden using steganographic techniques. Alternatively, the designer can encrypt the data, which will tend to remove any pattern from it. Additionally, if a user only desires to hide a small amount of data, the steganographic process can be designed so that hidden data perturbs only a few bits of a carrier file in which it is inserted. A further concealment technique is to process the image statistics so that they are the same as the original carrier file. Consequently, the statistics of the carrier file will be almost unchanged and it

becomes unlikely that the presence of the steganographic modifications can be detected mathematically.

A subverted employee of an organisation, or software masquerading as such an employee, may use steganography to try to pass sensitive information from inside the organisation's secure logical perimeter to an agent (human or software) outside that perimeter. Conversely, to protect an organisation's security a firewall or other barrier device can be provided to detect and potentially prohibit export of sensitive information. It may also be suspected that steganography is being used to conceal the transfer of sensitive information.

Products are available which attempt to detect instances of steganography, and they rely on statistical properties of images. To assess the effectiveness of these products, tests were carried out using them with a number of data samples: the samples were digitised images in which information was hidden using a range of steganographic tools freely available from the Internet and based on publicly known principles. The samples included one which was an intentionally poor example of steganography. With most data samples, the tests showed that available steganography detection products give too high a false positive rate when their sensitivity settings are set to give an acceptable false negative rate. Here "false positive" means apparent detection of steganography where none exists, and "false negative" means failure to detect an actual instance of steganography. This demonstrates the weakness of techniques that rely on statistical properties of images.

The present invention provides a method for detection of unwanted items stored in a computer system such as computer code, data or information characterised in that it comprises obtaining a signature consisting of at least part of a program capable of introducing unwanted items, the signature incorporating sufficient data to characterise the program adequately, comparing the signature with files stored on the computer system, and, if a match with the signature is found, providing an indication thereof.

The invention provides the advantage that it provides a means for detection of unwanted items that does not rely on analysing statistical properties of images.

The indication may incorporate an identification of at least one of:

a) an item responsible for the match,
b) the nature of the unwanted item,
c) the unwanted item's location in the computer system, and
d) the matching signature.

The unwanted items may be steganographic items.

The signature may be one of the following:

a) a continuous sequence of code taken from a program;
b) a continuous sequence of code taken from a program and beginning where that program begins,
c) a continuous sequence of code taken from a program but not more than 5% and not less than 0.167% of that program's code;
d) a kernel of a program;
e) a .DDL or .ocx file; and
f) code implementing both write to and read from a file.

Asserted file type may be ignored when comparing files with the signature. The step of comparing the signature with files may for each file be preceded by checking the respective real file type by reading the start of the file and excluding from comparing with the signature files having prearranged initial byte sequences.

Files not accessible by a system administrator may be excluded from comparison with the signature. Files compared with the signature may include logical wastebasket files, deleted files, compressed files, self-extracting executable files and polymorphic files.

No indication as aforesaid may be given in respect of some prearranged files despite their containing code which matches a signature.

In an alternative aspect, the present invention provides computer apparatus for detection of unwanted stored items such as computer code, data or information characterised in that the apparatus is programmed to:

a) obtain a signature consisting of at least part of a program capable of introducing unwanted items, the signature incorporating sufficient data to characterise the program adequately,

b) compare the signature with files stored on the computer apparatus, and

c) if a match with the signature is found, providing an indication thereof.

In a further alternative aspect, the present invention provides computer software for detection of unwanted items stored in a computer system such as computer code, data or information characterised in that the software contains instructions for controlling computer apparatus to obtain a signature consisting of at least part of a program capable of introducing unwanted items, the signature incorporating sufficient data to characterise the program adequately, to compare the signature with files stored on the computer apparatus, and, if a match with the signature is found, to provide an indication thereof.

The computer apparatus and computer software aspects of the invention may have preferred features equivalent to those of the method aspect of the invention.

In order that the invention might be more fully understood, embodiments thereof will now be described, by way of example only, with reference to the accompanying drawings, in which:-

Figure 1 is a flow diagram of a steganography detection process of the invention; and

Figure 2 is a flow diagram of a check file step which is incorporated in the Figure 1 process.

Referring to Figure 1, a flow diagram of a computer-implemented steganography detection process 10 is shown which starts at 12 and obtains a steganographic signature directory at 14 as will be described in more detail later. At 16, a check is made to see if there is any unread file in the directory, and if so at 18 a sample of length N bytes is read from the beginning of that file into a steganographic signature array directory. Here N is a positive integer chosen to be sufficiently large to avoid false positives as far as practically possible, and sufficiently small to allow the process 10 to detect use of variants of a steganographic program. In this example N was 500. The process 10 then iterates around steps 16 and 18 via a loop 20 until all files in the steganographic signature directory have been read and have provided respective N byte samples for the steganographic signature array directory.

The computer root search directory is accessed at 22 to enable all computer-stored files to be read. At 24 a check is made to see if this directory can be read: if not the error is

logged at 28, and a check is made at 30 to see if this directory is actually the root search directory. If it is the root search directory, the process 10 cannot proceed and ends at 32. If it is not the root search directory, at 34 a parent directory of the current directory is obtained and at 36 the parent directory is checked to see whether or not it has an unread subdirectory. The specific steganographic signature directory itself (see 14) is excluded from this procedure. If the parent directory has an unread subdirectory, at 38 the process 10 goes to that directory and it is checked at 24 to see whether or not it can be read. The procedure 24 to 38 then iterates until either the process 10 has ended at 32 or all readable subdirectories in the parent directory have been read. If the parent directory has no subdirectory which is unread but readable, at 40 the parent directory is checked to see whether or not it has an unread file. If it has an unread file, at 42 the file is compared with each signature file in the steganographic signature array directory (as will be described later in more detail), and the results of this comparison are stored. The procedure 40, 42 then iterates around a loop 44 until the parent directory has no unread file other than files for which it had been prearranged to exclude from checking for steganography. Excluded files comprised the system paging file and files which a system administrator could not examine: these were excluded to avoid a series of "no access" warnings.

At 46, another check is made at 30 to see if the parent directory is actually the root search directory: if not, a "parent of parent" directory is sought at 48, and via a loop 50 the procedure 24 to 46 iterates until it has either ended at 32 or the current directory has become the root search directory and all files in it have been checked for steganography. The results of checking at 42 are then reported at 52 to a user of the process 10, which then ends at 32. If the results indicate that any computer file contained data matching any steganographic signature, an unauthorised program on the computer being searched would have been identified. Once the unauthorised program is identified, appropriate remedial action can be taken by a computer system administrator.

Referring now to Figure 2, the check file step 42 is shown in more detail. It starts at 70, and at 72 and 74 respectively it checks whether or not the filename is on an exclusion list and need not be checked and file size is in accordance with predefined limits. If either the filename is on the exclusion list, or file size is not in accordance with predefined limits, at 76 the file is ignored and the check file step 42 ends at 78. If the filename is not on the exclusion list, and also file size is in accordance with predefined limits, then at 80 a sample of length N bytes is read from that file into a test signature file: it is the first N

bytes of the file which is read. A check is made at 82 for errors in this read process, and if there was an error the error is logged at 84 and the check file step 42 ends at 78. If there were no read errors, the steganographic signature array directory (see step 18 in Figure 1) is obtained at 86 and a steganographic signature is chosen from it at 88 to compare with the test signature in the file obtained at 80. This comparison is carried out at 90. If the steganographic signature does not match the test signature, at 92 a check is made to see whether or not there are any more signatures in the steganographic signature array directory. If there are any more signatures, via a loop 94 the procedure of steps 88, 90 and 92 iterates while steganography remains undetected until there are no more steganographic signatures left for comparison, in which case the check file step 42 ends at 78. If the steganographic signature does in fact match the test signature at 90, then at 96 the filename, the finding of steganography in it, the file location and the matched signature are recorded as results for reporting at 52 in Figure 1. The check file step 42 then ends at 78 until a new file is selected for checking at 40 and the step 42 iterates once more. Files identified as adulterated by steganography may be deleted or removed for investigation.

The steganographic signature directory at 14 was obtained by assembling a series of steganographic programs (two in the present example), these being programs which implement steganography as opposed to software, imagery or data which have become afflicted by it. Candidate steganographic programs were required to have at least 10,000 bytes and not more than 300,000 bytes, so N equal to 500 means that the signature represents not more than 5% and not less than 0.167% of the program it is taken from.

It is straightforward to obtain copies of most or all publicly available steganographic programs, as they are available free or obtainable by purchase, often from the Internet. In order to derive steganographic signatures, an important file in each of these steganographic programs was identified and chosen: wherever possible, this chosen file was the program's core steganographic kernel. For example, a steganographic program may comprise an ".exe" (execute) file to provide a computer user interface, and a ".DLL" (dynamically linked library) file to perform a computationally intensive computation when called by the .exe file. For such a program, the .DLL file was chosen to provide the N byte sample at 18 as it is liable to be smaller than the .exe file and more importantly less liable to be changed. A .DLL file is often a file which implements a mathematical function, and its kernel is that part of it which is essential to implement that function. Choosing a .DLL file also means that attempts to write a new steganographic program using it would

also be detected. It is best if the N byte sample is completely unique to the steganographic program to avoid false positives, and a .DLL file is more likely to be unique than an .exe file. Another possibility is a .ocx file which is similar to a .DDL file except that it has a different interface.

Once the chosen files were identified, they were put into a specific steganographic signature directory containing no other files. Access to this directory was banned to all users except administrators: this access control is available using well-known computer operating system facilities. The first N bytes of each of the chosen files was taken as the respective steganographic signature, because bytes at the beginning of a file are less likely to be altered than others. Each signature was then available for comparison with computer files at 90. The process 10 then compared these signatures with all files on the computer hard disk, apart from those in the specific steganographic signature directory itself and those on a prearranged exclusion list.

It was found that a wide range of values for "N" could be used for which no false positives occurred. Because all files on a computer hard disk other than the signature directory and those on the exclusion list are compared with steganographic signatures, a user of the process 10 can be confident that all steganographically modified files will be identified for which there were steganographic signatures in the signature directory obtained at 14.

The invention may alternatively be carried out by comparing each steganographic signature with each computer file asserted to be an executable file by an .exe suffix. This reduces the processing time. However, it is preferable to ignore the asserted file type when performing the process 10, in order to defeats attempts to hide unwanted steganographic software by changing the asserted file type. Moreover, it is possible to discern the real file type by reading the start of the file, and those files that start with initial byte sequences of known image and sound types can be eliminated from comparison in the process 10. The process 10 may be further enhanced by looking inside logical wastebasket files such as those found in common personal computers, or for deleted files to see if any steganographic files have been present on a computer in the past. It may be used with compressed files such as Zip files, cabinet files and TAR files. It may be used to look for self-extracting executable files and polymorphic files, in ways that are analogous to those used to detect some classes of computer virus.

The process of the invention may be speeded up by performing some tests more frequently than others. Alternatively or additionally, whenever a program is to be executed, a check may first be made that the program does not have a steganographic signature before it starts to execute. This check can be combined with a virus checking program, giving the possibility of a reduced time needed to perform both types of check. The checks may be made on all or some of the computers on a computer network

The steganographic signature may be an entire steganographic program, or as previously described it may be a subset of the program code such as the first N bytes of its kernel. Such a subset is a template indicating the relevant steganographic program. One advantage of the use of a template is that it may be faster to perform the process 10. A second advantage is that variants of a particular steganographic program may retain the template unchanged, which means that the variants need not be checked individually with different templates. A further advantage is that if a new version of a steganographic program is produced, there is a possibility that the template for the previous version may appear in the new version. This makes it possible that new unknown variants of steganographic software may be detected by the process 10. The signature or template should be sufficiently long, i.e. should contain a sufficient number of bytes, to characterise the steganographic program adequately to avoid significant numbers of false positives. It needs fewer bytes if it incorporates something highly specific to the steganographic program such as its kernel.

Although the invention has been described in terms of image carrier files and steganographic programs, it can be used for other types of carrier file and other types of unwanted items, i.e. computer code, data or information.

The process 10 is a reactive technique, in that it is necessary to obtain sample steganographic programs or templates derived from them. An alternative technique is to look for generic signatures. There are only a limited number of standard ways of accessing images in any given operating system. There are even fewer standard ways of writing images back to the computer's storage media. One method of identifying potential steganographic programs is look for programs that contain the necessary software calls to read and write images or other types of carrier file. Any file that contains both read and write software calls is a potential steganographic program. One alternative is to raise an alarm each time that this occurs. However, there some programs that would be difficult to use for steganography, such as Microsoft Paint, which are common on many computer

systems. The invention may use one or more white lists of acceptable files, that is files whose presence on a computer system is arranged not to give rise to an indication of such presence even though they contain code which if compared with a signature would indicate steganography. This avoids false positives arising from for example files such as graphics editors which contain both read and write software calls indicating a potential steganographic program, but which are not in fact actual steganographic programs.

The example 10 of a process of the invention was described for a situation where only what is referred to as a "logical hard disc" would be searched. In this connection, current operating systems may be unable to cope with hard discs of the size currently available, i.e. tens of gigabytes. In this case, a hard disc is divided up into regions (e.g. two gigabyte regions) each treated as a different drive (e.g. A, B, etc. drives) and accessed separately: these regions are referred to as logical hard discs. The process 10 operates only for a logical hard disc currently being accessed. It is possible to adapt the process 10 so that it automatically searches all a computer's logical hard discs, but as described a user must switch from disc to disc manually to do this.

Claims

1. A method for detection of unwanted items stored in a computer system such as computer code, data or information characterised in that it comprises obtaining a signature consisting of at least part of a program capable of introducing unwanted items, the signature incorporating sufficient data to characterise the program adequately, comparing the signature with files stored on the computer system, and, if a match with the signature is found, providing an indication thereof.

2. A method according to Claim 1 characterised in that the indication incorporates an identification of at least one of
   a) an item responsible for the match,
   b) the nature of the unwanted item,
   c) the unwanted item's location in the computer system, and
   d) the matching signature.

3. A method according to Claim 1 characterised in that the unwanted items are steganographic items.

4. A method according to Claim 1 characterised in that the signature is one of the following:
   a) a continuous sequence of code taken from a program;
   b) a continuous sequence of code taken from a program and beginning where that program begins,
   c) a continuous sequence of code taken from a program but not more than 5% and not less than 0.167% of that program's code;
   d) a kernel of a program;
   e) a .DDL or .ocx file; and
   f) code implementing both write to and read from a file.

5. A method according to Claim 1 characterised in that asserted file type is ignored when comparing files with the signature.

6. A method according to Claim 1 characterised in that the step of comparing the signature with files is for each file preceded by checking the respective real file

type by reading the start of the file and excluding files having prearranged initial byte sequences from comparing with the signature.

7. A method according to Claim 1 characterised in that files not accessible by a system administrator are excluded from comparison with the signature.

8. A method according to Claim 1 characterised in that files compared with the signature include logical wastebasket files deleted files and compressed files.

9. A method according to Claim 1 characterised in that files compared with the signature include self-extracting executable files and polymorphic files.

10. A method according to Claim 1 characterised in that in respect of some prearranged files no indication as aforesaid is given despite their containing code which matches a signature.

11. Computer apparatus for detection of unwanted stored items such as computer code, data or information characterised in that the apparatus is programmed to:
   a) obtain a signature consisting of at least part of a program capable of introducing unwanted items, the signature incorporating sufficient data to characterise the program adequately,
   b) compare the signature with files stored on the computer apparatus, and
   c) if a match with the signature is found, providing an indication thereof.

12. Apparatus according to Claim 11 characterised in that the indication incorporates an identification of at least one of:
   a) an item responsible for the match,
   b) the nature of the unwanted item,
   c) the unwanted item's location in the computer system, and
   d) the matching signature.

13. Apparatus according to Claim 11 characterised in that the unwanted items are steganographic items.

14. Apparatus according to Claim 11 characterised in that the signature is one of the following:

a) a continuous sequence of code taken from a program;
b) a continuous sequence of code taken from a program and beginning where that program begins,
c) a continuous sequence of code taken from a program but not more than 5% and not less than 0.167% of that program's code;
d) a kernel of a program;
e) a .DDL or .ocx file; and
f) code implementing both write to and read from a file.

15. Apparatus according to Claim 11 characterised in that it is programmed to ignore asserted file type when comparing files with the signature.

16. Apparatus according to Claim 11 characterised in that before comparing the signature with files it is programmed to check for each file the respective real file type by reading the start of the file and excluding files having prearranged initial byte sequences.

17. Apparatus according to Claim 11 characterised in that it is programmed to exclude from comparison with the signature, files not accessible by a system administrator.

18. Apparatus according to Claim 11 characterised in that it is programmed to compare with the signature logical wastebasket files deleted files and compressed files.

19. Apparatus according to Claim 11 characterised in that it is programmed to compare with the signature self-extracting executable files and polymorphic files.

20. Apparatus according to Claim 11 characterised in that it is programmed to give no indication as aforesaid in respect of some prearranged files despite their containing code which matches a signature.

21. Computer software for detection of unwanted items stored in a computer system such as computer code, data or information characterised in that the software contains instructions for controlling computer apparatus to obtain a signature consisting of at least part of a program capable of introducing unwanted items, the

signature incorporating sufficient data to characterise the program adequately, to compare the signature with files stored on the computer apparatus, and, if a match with the signature is found, to provide an indication thereof.

22. Computer software according to Claim 21 characterised in that the indication incorporates an identification of at least one of

a) an item responsible for the match,

b) the nature of the unwanted item,

c) the unwanted item's location in the computer system, and

d) the matching signature.

23. Computer software according to Claim 21 characterised in that the unwanted items are steganographic items.

24. Computer software according to Claim 21 characterised in that the signature is one of the following:

a) a continuous sequence of code taken from a program;

b) a continuous sequence of code taken from a program and beginning where that program begins,

c) a continuous sequence of code taken from a program but not more than 5% and not less than 0.167% of that program's code;

d) a kernel of a program;

e) a .DDL or .ocx file; and

f) code implementing both write to and read from a file.

25. Computer software according to Claim 21 characterised in that it contains instructions for controlling computer apparatus to ignore asserted file type when comparing files with the signature.

26. Computer software according to Claim 21 characterised in that it contains instructions for controlling computer apparatus to check real file type by reading the start of the file and exclude files having prearranged initial byte sequences from comparison with the signature, and to do so prior to comparing the signature with files.

27. Computer software according to Claim 21 characterised in that it contains instructions for controlling computer apparatus exclude from comparison with the signature files not accessible by a system administrator.

28. Computer software according to Claim 21 characterised in that it contains instructions for controlling computer apparatus to compare with the signature logical wastebasket files, deleted files and compressed files.

29. Computer software according to Claim 21 characterised in that it contains instructions for controlling computer apparatus to compare with the signature self-extracting executable files and polymorphic files.

30. Computer software according to Claim 21 characterised in that in its instructions provide computer apparatus controlled by it to give no indication as aforesaid in respect of some prearranged files despite their containing code which matches a signature.

Abstract

Detection of unwanted items stored in a computer system such as computer code, data or information includes obtaining signatures characteristic of programs of interest such as steganographic programs. A signature may consist of an initial 500 bytes from a .DDL file taken from the program's core steganographic kernel. Each signature is compared with files on the computer system other than files on a prearranged exclusion list. If a signature is found to match data in a file, the filename, the finding of steganography in it, the file location and the matched signature are recorded for output to a system user.

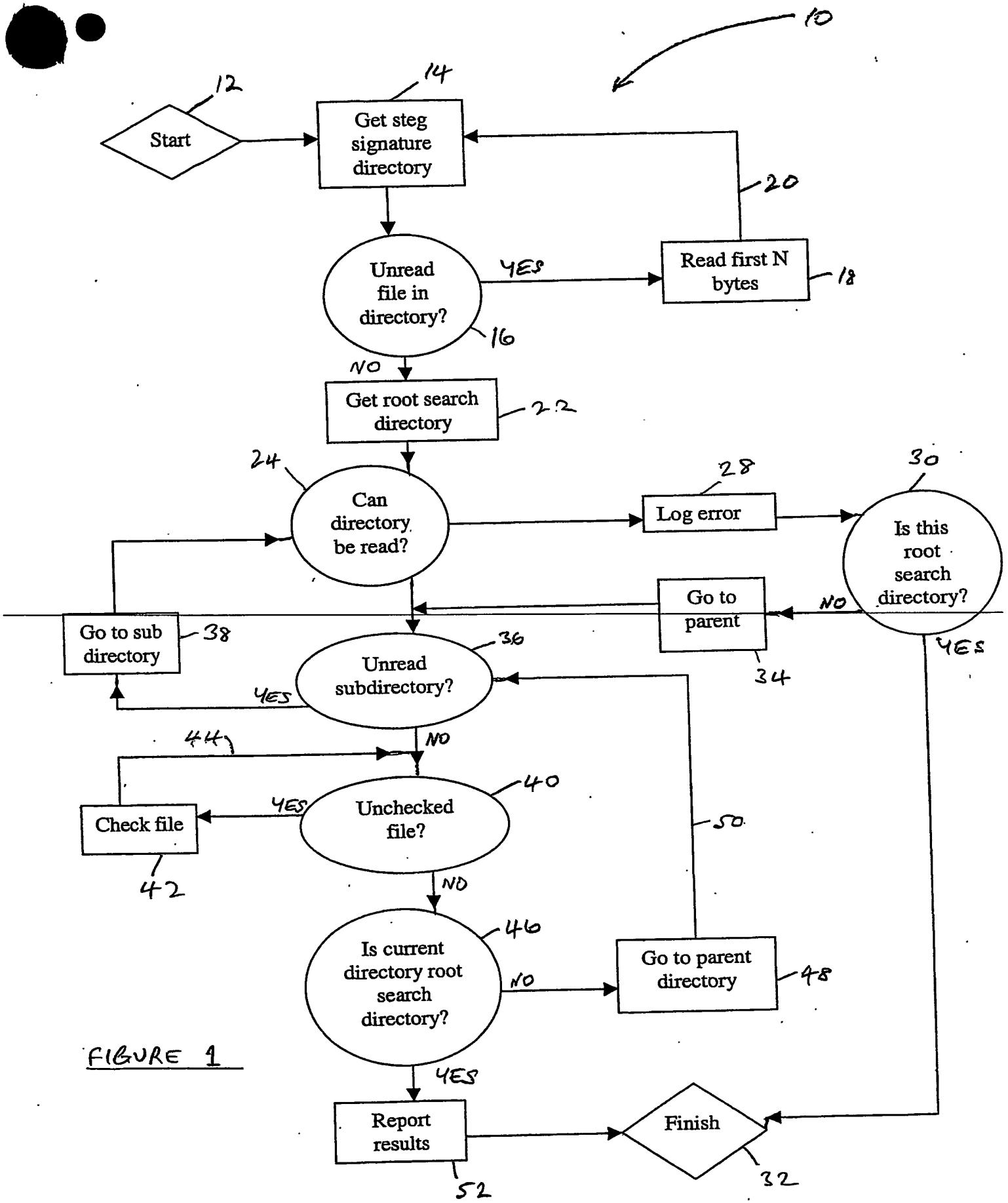
10
12
Start
14
Get steg signature directory
20
Unread file in directory?
YES
Read first N bytes
18
16
NO
Get root search directory
22
24
Can directory be read?
28
Log error
30
Is this root search directory?
NO
YES
Go to parent
34
Go to sub directory
38
36
Unread subdirectory?
YES
NO
44
40
Unchecked file?
YES
Check file
42
50
NO
46
Is current directory root search directory?
NO
Go to parent directory
48
YES
Report results
52
Finish
32

FIGURE 1

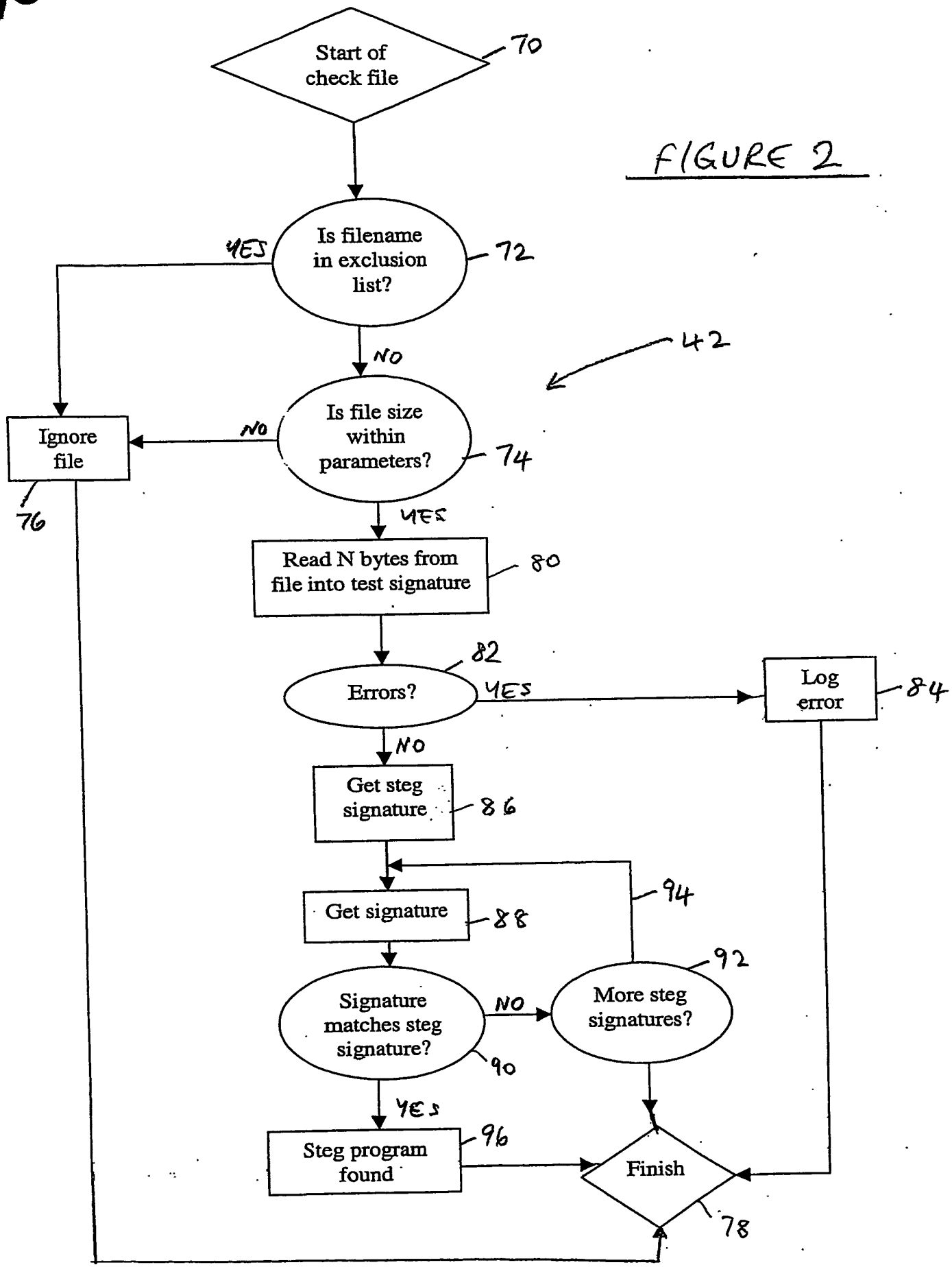
Start of check file
70
Is filename in exclusion list?
YES
72
NO
42
Is file size within parameters?
NO
74
Ignore file
76
YES
Read N bytes from file into test signature
80
82
Errors?
YES
Log error
84
NO
Get steg signature
86
94
Get signature
88
92
Signature matches steg signature?
NO
More steg signatures?
90
YES
Steg program found
96
Finish
78

FIGURE 2

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/GB04/004510

International filing date: 25 October 2004 (25.10.2004)

Document type: Certified copy of priority document

Document details: Country/Office: GB
Number: 0325767.2
Filing date: 05 November 2003 (05.11.2003)

Date of receipt at the International Bureau: 19 November 2004 (19.11.2004)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)